Functionality or User Interface: Which is easier to learn when changed?

Himanshu Zade
International Institute of Information
Technology - Hyderabad, India 500032
Email: himanshu.zade@research.iiit.ac.in

Venkatesh Choppella
International Institute of Information
Technology - Hyderabad, India 500032
Email: venkatesh.choppella@iiit.ac.in

***Abstract*—The rapid release model of software introduces frequent updates to the existing software every twelve-eighteen weeks, forcing a user to get accustomed to its new features. We propose an experimental study to compare the learning gaps that are introduced when a user is exposed to a new version of a software, with which he is previously acquainted. In order to explore the problem, we propose four models of a machine, with each model involving an update either to the functionality, or to the user interface, or both. We conducted a between-subjects experimental study with thirty-two participants who performed two tasks successively on two models of a machine, the second one being a updated model of the first. The analysis of the data using ANOVA implies that a change in the user interface dominates a change in the functionality. Results indicate that 88% of the errors were caused due to a change in the user interface. 87.5% of the users who underwent a change in the user interface hold this change responsible for the learning gap, while only 56.25% users who underwent a change in the functionality consider it to be a potential reason for the learning gap.**

***Index Terms*—User Interfaces, Human computer interaction, Software metrics**

I. Introduction

Understanding the factors responsible for software to be successful in the market is both important and crucial. Modification of the existing software has always been one such important factor that necessitates a user to adapt to the change. A change from a previous system to a target system can happen through four important adoption techniques: Plunge, Parallel, Pilot and Phased [1]. The phased adoption techniques have made it possible to have radically higher number of software releases. This gave rise to Rapid Release Models that enables a new version of a software to be released to a user within twelve to eighteen weeks [2]. This has enforced the developers to be more competitive and adopt the Rapid Release Model, leading to more and frequent releases of newer versions of the same software.

With the introduction of the newer versions of a software in market, users are exposed to a new level of complexity. This complexity is a result of the variations in the features of the software to which the user is not accustomed. Failure to address the learning gap, thus introduced, can be harmful for the market acceptance of the software. For example, Mozilla Firefox browser lost its audiences due to its rapid release cycle [3] . Such variations can be brought about by addition, subtraction or modification of the features of the software, either to the functionality or to the user interface. In order to enable the user to adapt to these changes in minimal time and effort, it is essential to study how can the impact of this complexity on required user learning be minimised.

The prior experience of the user with a software significantly affects the way a user interacts with it [4]. This experience, however, conversely influences and impacts the process of adapting to the variations introduced within the software. We scope our problem to the changes brought about by rapid releases using a phased adoption. Within phased adoption, we consider a change introduced at each phase either to the functionality of the system, or to the interface, or both. We try to explore how each of these changes introduce a learning gap in between the user understanding about the current software and the new software. For this study, we developed a machine that acts as a metaphor for a software. We define this machine as a process that takes user input, and outputs the corresponding behavior. We consider four models of the machine having variations within the functionality, and the user interfaces. Each model corresponds to a different version of the software. A comparative study between the different models of this machine is performed in order to investigate the learning gaps introduced while modifying the machine model. We performed two-factor ANOVA to identify whether the effect produced is a main effect due to any one of the two factors or interference effect involving both the factors. This was followed by a one-way ANOVA to verify if the means of the different groups are significantly different. The study concludes that a change in the functionality is easier to learn as compared to a change in the user interfaces.

II. Background

Research to identify parameters that drive software successful helps the developer to design better software. User attitude and resistance to change is identified as an important factor that drives the success of a software [5]. Change can be introduced in existing software in many ways. Studies pertaining to change management have shown that out of these several potential options, there exists a pathway that allows the least resistance for the user to adapt to the changes. Such a path of least resistance finds many applications and has been well investigated in relation to information systems [6], usability and security [7], and probability and stochastics [8].

Table I
2X2 FACTORIAL DESIGN OF THE EXPERIMENT TO COMPARE THE LEARNING GAPS, WHEN SWITCHING TO NEW USER INTERFACE OR NEW FUNCTIONALITY

Transition of User Interface (column variable) Transition of Function (row variable)	**Regular keypad with 9 buttons**	**New keypad with 5 buttons**
Enter a ten digit number	A	B
Traverse a graph, from blue point to red point	C	D

Similar notions of user resistance to change have been studied in the resistance theory [9]. In this theory, resistance has three perspectives: people oriented, system oriented and interaction oriented. The system oriented resistance deals with how resistance is introduced due to the characteristics of a system design or the user interface. A significant body of literature concerned with designing systems for usability has emerged in context to system oriented resistance [10]. It, however, concentrates only on user interfaces, and does not involve the changes in the functionality as a potential reason to affect usability. We, therefore, believe that our study is unique as it involves a cross comparison of the effect of change in functionality or user interface on the learning gap introduced while switching to new machines.

III. Problem and Approach

The aim of the study is to compare the learning gaps introduced when the user switches to a machine with a new interface as against a machine with a new functionality. In other words, we try to identify the factor that drives the learning gap: a change in the functionality, or a change in the user interface. To study the impact of all possible combinations of the transitions from the existing to new, functionality and user interfaces, we conducted an experiment having 2X2 Factorial Design. Table I shows the four possible combinations: A, B, C, and D, considered for the study. We will describe the design in detail in Section IV-A.

Our driving question, thus, is which factor results in greater difficulty for a user to get accustomed to a newer model of the machine. The four possible combinations of varying functionality and user interface can be realised through the four models A, B, C and D, a machine. We try to identify the factor that affects the ability of a user to use the machine efficiently, his knowledge about the new functions in the machine, or how to use these functions from the user interface. Identifying the more crucial parameter in the above two proposed concerns will help us to ease the difficulty posed in successful adaptation of the new machine model [11]. Our study, thus, addresses the following research questions:

RQ1-Which of the two, introduction of a new user interface or introduction of a new functionality, is more likely to affect the user's understanding about the behavior of the machine model, based upon the objective assessment of the user performance evaluated against the allotted tasks?

RQ2-Which of the two, introduction of a new user interface or introduction of a new functionality, poses a larger potential gap in realising the revised behavior of the machine model, based upon the subjective assessment of the user towards the transition he is exposed to?

RQ3-Is there a relative difference in the degree to which the user is able to complete the task across different models of the machine?

RQ4-Is there a relative difference in the number of errors committed by a user when switching from one model of the machine to another?

We have proposed a corresponding measure for each research question raised above in Section V, and answered the questions in Section VI.

IV. Experimental Design

In this section, we describe our experimental setup used to explore the switching across different models of a machine. The aim of the experiment is to explore the potential learning gaps introduced when switching to newer models of a machine. We classify the switch in the two models of a machine as a change in the functionality, or a change in the user interface. We, therefore, choose the functionality of the machine and its user interface as the two independent variables. Time taken by the user to finish the task allotted to him, and the number of unnecessary clicks performed by the user, account for the two dependent variables. Time taken, here, is a measure of the user performance and hence his understanding of the machine. On the other hand, number of unnecessary clicks measures the errors committed by the user. These two, together, help us measure the user understanding of the machine model.

For our study, we consider the following two hypothesis-

1) H0 (Null Hypothesis)- The overall learning gap introduced when switching to a new model of a machine remains unaffected whether the change is due to a new feature or a new user interface.
2) H1 (Alternate Hypothesis)- The overall learning gap introduced when switching to a new model of a machine is more if the user interface is changed as against to introduction of a new feature.

A. Apparatus

The entire experiment was conducted on a machine with its four models being tested, each, upon a different group of participants. The four models of the machine A, B, C, and D, shown in Figure 1, Figure 2, Figure 3, and Figure 4, respectively are implemented using the Processing Language[1] used to create images, animations, and interactions.

In order to realise the aim of our study, we model the different versions of a software as the four models of a machine. The four models A, B, C and D, are a metaphor for the different versions of software. The model A (Figure 1) represents the current version with which the user is well acquainted. We

[1] http://processing.org/

Figure 1. Model A consisting of a known functionality and a well acquainted user interface

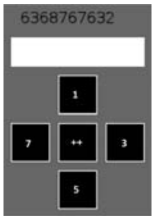

Figure 2. Model B consisting of a known functionality and a new user interface

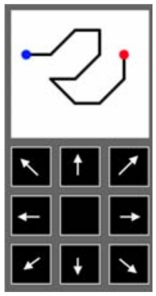

Figure 3. Model C consisting of a new functionality and a well acquainted user interface

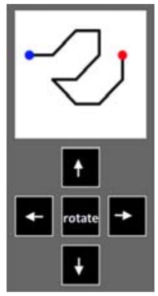

Figure 4. Model D consisting of a new functionality and a new user interface

introduce a change in the feature of this machine, either to the functionality leading to model C (Figure 3), or, to the user interface leading to model B (Figure 2). This introduction of change results in gap for a user to learn, referred to as the learning gap. Our study aims to compare this learning gap introduced when switching between different models of the same machine.

We aim to perform a comparative study across all the possible combinations of transitions. We vary the first factor, functionality, across two levels: functionality that a user is well acquainted with, and, a functionality that is new to the user. For example, we propose a new model D (Figure 4) of the same machine such that it varies from model B in terms of functionality. Similarly, we vary the second factor, user interface, across two levels based on the experience that a user has with the user interface. For example, model D is different from model C in terms of user interface. Such a modelling allows us to study the effect of varying both the independent variables, one at a time, using a 2X2 factorial design.

In model A, a single key press displays the corresponding number on the screen. This model represents the typical scenario, for example when a user enters a phone number consisting of digits 1-9 using a nine button standard mobile interface. Model B allows the user to accomplish similar task as of model A using the five button interface. The top, right, bottom and left buttons when pressed, output digits 1, 3, 5 and 7 respectively. For the even digits, user needs to press the key corresponding to the odd digit immediate before, followed by the center button. For example, the top button followed by center button outputs digit 2. By pressing the center button once more, however, neither 3, nor any output is generated. Its limitation to output digit 9 does not interfere with the goal of the study since the task (TableII) to generate the given number did not involve the digit 9. Model C maps each button, but the center button, to a direction indicating an arrow, originating from the center button towards the pressed button. Each button thus helps the user traverse one edge of the graph. For example, in order to traverse towards right in the graph, the user needs to press the button situated towards the right of the center button. No observable output is generated if the user performs an undesirable click with respect to the allotted task in which he is to move the blue dot towards the red dot as directed in the graph. For example, pressing the button situated towards the bottom of the center button will not generate any output for the previous example. The behavior of top, right, bottom, and left buttons on the user interface of Model D follows from that of Model C. The center button when pressed following any other button indicates a clockwise rotation of 45 degrees to the direction specified by the preceding button. The top button, thus, when followed by center button, assists the user in traversing an edge of the graph that points toward the top right corner. Successively pressing the center button once again, however, does not result into any observable output.

Next, we explain how the four models of the machine are comparable on the grounds of user effort. We realise each machine as a black-box that takes some input and outputs the corresponding behavior. The output is a sequence of either digits or graph edges. The input to each of the models is a key press. Let us denote a key press, by a user, as k. The trace of key presses required to observe a single digit or traverse a graph edge, is a sequence s. A sequence $s = \{k_1, k_2\}$, thus, denotes two keys pressed consecutively such that, the sequence s results into an observable output. The notation k_i denotes i^{th} key press in a sequence.

For machine model A, number of key presses required to output a single digit is 1. Thus, the sequence s for machine A is $s = \{k_1\}$. Similarly, for machine B, the key presses required to output a single digit depends on whether the digit is even or odd. Thus, for 7, sequence t is $\{k_1\}$ where as for 8, the sequence is $s = \{k_1, k_2\}$ because one has to press 7 and the incrementing key. The sequences for machine C and D remain same as that of A and B respectively as they differ only in their set of outputs.

Regarding the familiarity with the interfaces of machines A and B, the effort to learn the interface A is less, given the experience of the user with the conventional number-pad interface. Thus, the tasks appears to be cognitively less complex. Since the user has no prior experience with the interface of the machine model B, the tasks appears to be cognitively more complex. Thus, more user effort is required in learning the interface for model B.

This effort is more when there is a requirement of two key presses for a single output and might result in twice the time being consumed to achieve the same output as compared to using interface A. The invalidation of this assumption is supported by the results of the study, discussed in Section VI.

B. Procedure

The study involved forty participants at the beginning. All the participants were computer science graduate students with their age varying from eighteen to twenty-three years with an average age of twenty years. Out of forty, eleven participants were females and twenty-nine were male. Each of them had more than seven years of experience of using a mobile phone and calculator thus confirming their well developed understanding about using a number-pad to enter digits. They

Table II
THE TASKS PERFORMED BY EACH PARTICIPANT BELONGING TO ONE OF THE FOUR GROUPS. THE CORRESPONDING TRANSITION OF FUNCTIONALITY OR USER INTERFACE ENCOUNTERED WHEN SWITCHING BETWEEN TWO MODELS OF THE MACHINE IS MENTIONED ALONGSIDE.

Grp #	**Switch between two models**	**Transition encountered**	**Task 1**	**Task 2**
1	A–B	User Interface	Enter the ten digit number displayed to the user using the nine button interface on model A	Enter the ten digit number displayed to the user using the five button interface on model B
2	C–D	User Interface	Move the blue dot towards the red dot by traversing the graph using the nine button interface on model C	Move the blue dot towards the red dot by traversing the graph using the five button interface on model D
3	A–C	Functionality	Enter the ten digit number displayed to the user using the nine button interface on model A	Move the blue dot towards the red dot by traversing the graph using the nine button interface on model C
4	B–D	Functionality	Enter the ten digit number displayed to the user using the five button interface on model B	Move the blue dot towards the red dot by traversing the graph using the five button interface on model D

were neither exposed to the new functionality of traversing graphs, nor the new user interface, before the conduct of the experiment.

We followed a between-group design, and divided the forty participants into four groups of ten each. Participants from each of the four groups were to perform a set of two tasks each, such that they experience a change of any one factor, either of functionality or of user interface. A detailed description of tasks, and the group-wise switching between two models of the machine can be seen in Table II. The entire study took about fifteen minutes for each user. Every user was given incentives as a reward for contributing to the study in terms of participation.

Before the start of the study, we took the demographics of all the participants. The main study was carried in three phases: (a)Task 1, (b)Task 2 and (c)Qualitative feedback. Before beginning with the user tasks, we gave the users instructions on how to perform the user study. Following this, the user performed the two tasks that he was allotted. Later, each user gave a qualitative feedback about the study.

We gave the user a set of instructions explaining the functionality and the behavior of the user interface of the particular model that the user was allotted for performing the task. Following these instructions, we performed a small task before the user. The sample demonstration ensured that all the users had a similar level of understanding regarding how the model works. Before making the user actually perform any task, his doubts, if any were clarified. After educating the user about the task, the user performed his first task. No interaction with the user was entertained while he performed the task. Post the task, the user had to answer a set of subjective questions pertaining to whether he understood the model and the allotted task correctly, and to address any issues that he faced to complete the task.

After finishing with the first task on one model of the machine, the user was now exposed to the later model of the machine that varied in one factor: function or interface from the previous model. We gave a similar treatment to the user as previously for the previous model.

The two tasks were followed by a subjective interrogation with the user that began with the user think aloud about any difficulties he faced while switching between the two models of the machine. We enquired if the user found out any difference in completing the allotted tasks on any or both of the models. If he found any difference between the two models, we further asked him what differentiated the two models of the machine. Participants of groups 1 and 2, who performed the task on model B and D, were specially questioned how the introduction of a center button affected their experience. Towards the end of the experiment, each user, depending on his interest, was explained about the study in further detail and how it helps address the goal of the study.

V. RESULTS

As the user accomplished the task, we recorded the amount of time taken by the user to finish the task, the number of unnecessary clicks he performed, and the extent to which the user completed the task. We, thus, measured the following properties because they are well known and commonly suggested measurements for evaluating overall learning of devices. The first three are referred to as the usability measurement criteria in many other works [12].

- M1- Efficiency: It measures the resources consumed by the user to complete the task. In the case of our study, time taken by the user to accomplish a task gives this measure.
- M2- User Satisfaction: It is a subjective measure of the comfort and subjective acceptance of the model to the user.
- M3- Effectiveness: It is the level of completeness of an action. In our case, suppose a user was able to perform nine of the ten steps, we account the effectiveness as 90%.
- M4- Accuracy: It considers the total number of errors committed by the user in performing the task. An increase in the number of undesirable clicks, here, reduces the accuracy.

The measure of each property mentioned above, answers the corresponding research question raised in Section III. For example, M1 answers RQ1 by measuring time, and thus,

Table III
MEANS CALCULATED FOR EACH GROUP OF SIXTEEN PARTICIPANTS. F1 AND F2 INDICATE THE TWO LEVELS OF FUNCTIONALITY, WHILE UI1 AND UI2 INDICATE THE TWO LEVELS OF USER INTERFACE.

Levels of the varying factor	UI1	UI2	Total
F1	13.876	27.156	20.515
F2	17.487	33.987	25.737
Total	15.681	30.571	23.126

Table IV
TWO-FACTOR ANOVA RESULTS DESCRIBING THE EFFECT OF A CHANGE IN ONE OR BOTH THE FACTORS: FUNCTIONALITY AND USER INTERFACE

Factor responsible for the effect observed	F	p
Functionality only	10.657	0.005226
User Interface only	33.8651	<0.0001
Functionality and User Interface both	2.014	0.176461

evaluates the user performance. A detailed discussion follows in Section VI. In order to ensure normal distribution of data, we discard two outliers from each group. For the rest of the study, we consider and analyse the remaining thirty-two users. The Figure 5, Figure 6, Figure 7, and Figure 8, display the performance of these thirty-two users through visualisation of the measure M1. These four plots correspond to the group treatment 1, 2, 3 and 4 respectively (Table II). The plots map the user on X axis to the time taken by him to finish the task on Y axis. The two lines indicate the performance of the user on the two different models of the machine.

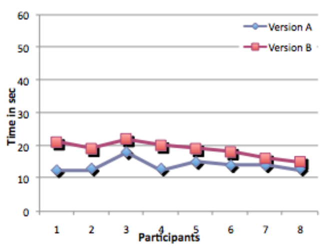

Figure 5. Performance Plots for AB transition

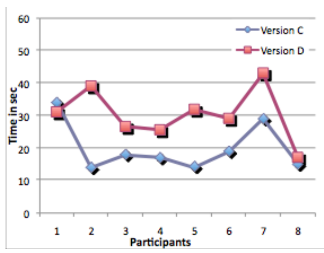

Figure 6. Performance Plots for CD transition

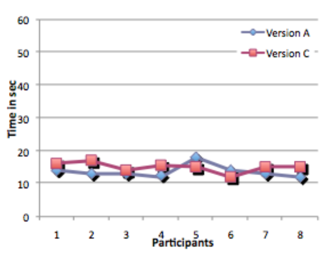

Figure 7. Performance Plots for AC transition

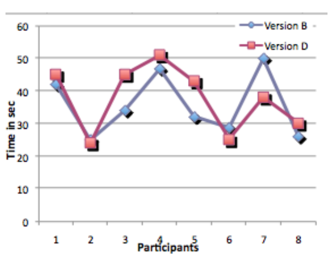

Figure 8. Performance Plots for BD transition

A. Statistical Analysis

The study aims to determine the factor, functionality or user interface, or both, responsible when the user acquaints himself to a new model of the machine. To realise this across two levels (well known and new) of both the factors (functionality and user interface), we chose a 2X2 factorial design (Table I). We conduct a two-factor ANOVA test on the data, to figure out if the change in the performance of the user is a main effect introduced due to any of the dominating factors, or an interaction effect due to equal involvement of both the factors.

Each transition AB, CD, AC and BD across different models of the machine was effectively tested upon eight users. We, thus, have a set of sixteen users who performed the allotted tasks using the same model. We group the performances of such users under one group, resulting into four groups having performance measures of sixteen users each. The means for each of the groups with sixteen participants each, are described in Table III. We now conduct the two-factor ANOVA to get the results. A summary of the results is shown in Table IV.

VI. DISCUSSION

The analysis was designed to assess the effects of both, change in functionality and in interface, on learning gap introduced for a user to adapt to the newer machine model. Analysis comprised a 2 (functionality: known vs. unknown) * 2 (user interface: known vs. unknown) between-participants design, with the time required to perform the task as the dependent variable. This was evaluated against measure M1. The two-factor ANOVA analysis (Table IV) revealed no significant main effect for functionality change. However, a significant main effect for user interface variation was observed. We did not find any sound interaction effect between both the factors. This is clearly evident in Figure 9 and Figure 10.

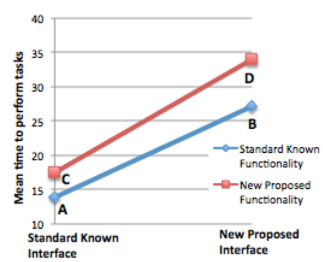

Figure 9. Two-factor ANOVA results indicating learning gap due to change in the functionality

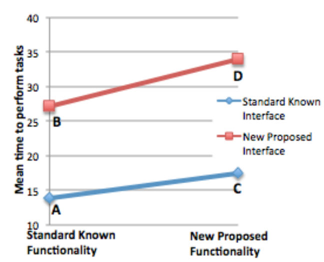

Figure 10. Two-factor ANOVA results indicating learning gap due to change in the user interface

The two-factor ANOVA does not provide information whether specific means are significantly different from one another; it only says that there exists no significant interaction between the two factors. To address this, a one-way ANOVA was computed on each group of sixteen participants. The analysis revealed a significant overall one-way effect (F statistic = 21.47, $p < 0.05$). Pairwise comparison of sample means via Turkey HSD test indicated no significant variation between models A and C, and between models B and D. This further supports our finding.

From the analysis, we therefore, conclude that user interface is the dominant factor of the potential two factors: functionality and user interface, explored within the scope of the study and thereby answer RQ1. We, therefore, accept H1 and reject H0. In order to answer RQ2, we performed a subjective analysis of user satisfaction via measure M2. It indicated that out of the 16 users exposed to change in the user interface, 14 users, that is, 87.5% of them accepted that change in the interface was the major difficulty while switching across the models. This figure was considerably more as compared to 56.25% users (9 out of 16) admitting functionality change to be the factor causing difficulty while switching across the models. Since all the participants were able to complete their tasks, no significant results were produced for measure M3, that is, the degree of completeness of the allotted task. We suspect the simplicity of the chosen tasks and participants with a high technical expertise to be the prime reasons for our failure to affirm RQ3 positively. The distribution of errors across the four models of the machines: A (3), B (17), C (6) and D (49) indicated that change in interface accounted for 88% of the committed errors. The measure M4, thus, addresses the concern of RQ4 and supports our finding that change in the user interface is more crucial for the user to adapt.

We had anticipated in Section IV-A that the user effort is more when there is a requirement of two key presses for a single output and might result in twice the time being consumed to achieve the same output through interface of model B or D as compared to using interface of model A or C. This assumption was invalidated during the study through quantitative and qualitative measures. The time lag required to press k_2 after k_1 came out to be five percent of the total time taken for pressing both the keys in sequence $\{k_1, k_2\}$. Also, on subjective analysis, the user replied that it was deciding the first key k_1 that required thought followed by k_2 implicitly. This finding supports the idea that both the interfaces, and hence all the models vary only on the factor of experience, each user has with the particular machine model.

We kept the scale of experiments small to limit the variations introduced stepwise in each of the model strictly to either of the factors- functionality or the interface. Another factor responsible in limiting the scale is the difficulty to ensure that the same machine allows variations to formulate models analogous to the four models considered for our study. Despite of the small scale, the adequacy of the experiments follows from the tight coupling between the four models as they evolve from the same machine. Moreover, we conducted the study in a manner that allows us to measure exactly one factor through each transition that a participant group underwent. A good way of scaling it up can be to consider more tasks within each model, that can be completed through both the interfaces, to yield a stronger conclusion. Another limitation is the confinement of the study to users that are well experienced with a machine model before coming across any variation in the two factors under consideration. Having participants with varied levels of experience can also be insightful.

VII. Conclusion

The study set out to compare the learning gaps that are introduced when a user is exposed to a new version of a software, varying in terms of functionality, or user interface, or both. We make two important contributions through our experimental study. The first is that it allows modelling of different versions of a software, varying in functionality and user interface, through the four proposed models of the machine. The study also experimentally proves that the overall learning gap introduced when switching to a new model of a machine is more if the user interface is changed as against to introduction of a new functionality. We see, thus, that irrespective of the acquaintance with the functionality offered by the device, change of interface is a crucial factor to be considered for making the release of a newer version of software successful.

Acknowledgment

The authors would like to thank Harini Sampath, instructor for the Research Methodologies elective offered at IIIT-Hyderabad, for the timely guidance in shaping up the study.

References

[1] Artur, "Developing business/it solutions," [Online; accessed on September 2012]. [Online]. Available: http://www.scribd.com/doc/396849/Developing-BusinessIT-Solutions

[2] "The rapid release model for software development," [Online; accessed on 20-09-12]. [Online]. Available: http://www.realsoftware.com/support/rapidrelease.php

[3] B. Jono, "Everybody hates firefox updates," [Online; accessed on 20-09-12]. [Online]. Available: http://evilbrainjono.net/blog?permalink=1094

[4] Y. Kitamura, Y. Yamaguchi, I. Hiroshi, F. Kishino, and M. Kawato, "Things happening in the brain while humans learn to use new tools," in *Proceedings of the SIGCHI conference on Human factors in computing systems*, ser. CHI '03. New York, NY, USA: ACM, 2003, pp. 417–424. [Online]. Available: http://doi.acm.org/10.1145/642611.642684

[5] H. Barki and S. L. Huff, "Change, attitude to change, and decision support system success," *Information and Management*, vol. 9, no. 5, pp. 261–268, 1985. [Online]. Available: http://www.sciencedirect.com/science/article/pii/0378720685900503

[6] R. Hirschheim and M. Newman, "Information systems and user resistance: Theory and practice," *The Computer Journal*, vol. 31, no. 5, pp. 398–408, 1988. [Online]. Available: http://comjnl.oxfordjournals.org/content/31/5/398.abstract

[7] K.-P. Yee, "User interaction design for secure systems," in *Information and Communications Security*, ser. Lecture Notes in Computer Science, R. Deng, F. Bao, J. Zhou, and S. Qing, Eds. Springer Berlin / Heidelberg, 2002, vol. 2513, pp. 278–290. [Online]. Available: http://dx.doi.org/10.1007/3-540-36159-6_24

[8] H. P. Young, "The evolution of conventions," *Econometrica*, vol. 61, no. 1, pp. 57–84, Jan. 1993, [Online; accessed on September, 2012]. [Online]. Available: http://www.jstor.org/stable/2951778

[9] J. J. Jiang, W. A. Muhanna, and G. Klein, "User resistance and strategies for promoting acceptance across system types," *Information and Management*, vol. 37, no. 1, pp. 25 – 36, 2000. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0378720699000324

[10] B. Shneiderman, *Designing the User Interface: Strategies for Effective Human-Computer Interaction*, 3rd ed. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1997.

[11] D. Brion, "Davy brion gave a interesting perspective on what does it take to learn new things." March 2012. [Online]. Available: http://davybrion.com/blog/2012/03/thoughts-on-learning-new-things/

[12] A. F. Yamashita, W. Barendregt, and M. Fjeld, "Exploring potential usability gaps when switching mobile phones: an empirical study," in *Proceedings of the 21st British HCI Group Annual Conference on People and Computers*, ser. BCS-HCI '07, vol. 1. Swinton, UK, UK: British Computer Society, 2007, pp. 109–116. [Online]. Available: http://dl.acm.org/citation.cfm?id=1531294.1531309